INFORMATYKA EKONOMICZNA BUSINESS INFORMATICS 2(24) • 2012
ISSN 1507-3858

Stanisław Stanek
Tadeusz Kosciuszko Land Forces Military Academy
e-mail: stan_stanek@neostrada.pl

Mariusz Żytniewski
Katowice University of Economics
e-mail: mariusz.zytniewski@ue.katowice.pl

MICROFORMATS IN SOFTWARE AGENT DEVELOPMENT

Abstract: The article is focused on issues concerning the development of the semantic Web and the possibility of using microformats in software agent technologies. The first section provides an introduction to the information society and Web 2.0. The second section outlines the concepts of the semantic Web and microformats, and presents a sample application of the JENA library. The final section delivers conclusions.

Keywords: semantic Internet, microformats, software agent.

1. Introduction

Now that the Internet has become the primary medium for information interchange, the information society (being the subsequent stage of development following the industrial society) is emerging around the popular need to transfer, process and store information. As Zieliński [2008] puts it, this type of society is governed by the following principles:

- production of information – mass generation of information in response to mass demand and to enable mass retrieval and use of information,
- storage of information – the technical means that make it possible to accumulate and store infinite amounts of information,
- processing information – the development of technologies and standards in an effort to establish common descriptors of information and information exchange methods,
- sharing information – the provision of information regardless of time and space,
- retrieval of information – making sure that everyone who wants to retrieve an item of information can receive it,

- use of information – universal and unlimited access to the Internet as a source of information.

Software agents are a good example of technology designed to support human activity and making use of the same mechanisms of knowledge representation and processing that are commonly found throughout the Internet. To be able to operate efficiently, though, such technologies call for a shift away from representing Internet resources in the HMTL format toward solutions based on Web services and ontologies, which is an intrinsic feature of the semantic Internet or Web 3.0. With the emergence of Web 3.0, semantic methods of knowledge processing – identified with such standards as, e.g., RDF, RDFS, OWL, OWL2 – can be employed.

As these standards are embraced, it becomes feasible to use distributed and autonomous mobile entities termed by computer science as software agents. An added value arising from the application of software agents can be attributed to some of their unique properties, such as autonomy, the capability of collaboration and exchanging information within a community and the ability to cognize the environment.

The autonomous agent is defined as a system placed within and being part of an environment, capable of analyzing and influencing that environment over time as well as of acting toward the achievement of specific goals and simulating the impact of changes in that environment [Luck, Ashri, d'Inverno 2004]. It should be seen as a piece of software characterized by, e.g., proactivity, autonomy, mobility, openness, personification and personality [Stanek 2007]. If equipped with artificial intelligence, such systems can support humans or even replace them.

Groups of software agents can be regarded as multi-agent systems. According to Fasli [2007], the principal advantage of agent technologies is that they allows the development of complex and distributed systems composed of simple basic entities known as agents. In such systems, agents representing individuals and institutions act within a common environment, each performing its tasks.

Before this can occur, however, adequate mechanisms to represent software agents' knowledge have to be in place. One array of such solutions supporting the semantic representation of agents' knowledge is ontology languages developed within the framework of the semantic Web.

2. The semantic Web and microformats

The above mentioned concept of information society relates to an idea embedded in Web 2.0, proposing that Internet users should be assumed to be able to edit and publish content owing to the availability of platforms facilitating the creation of community portals. Such portals enable users to actively contribute to the dissemination of information, boosting their sense of membership in a specific group.

Modern technology offers multiple ways of publishing information, e.g., via RSS channels or open APIs as well as of searching and retrieving it by dedicated software applications, such as software agents. Information contained in a website can become a knowledge resource usable by computer software if it is represented in a certain way. Therefore, the operation of such software (agents) usually relies on the idea of representing knowledge in the form of ontologies and microformats.

2.1. The semantic Web

The semantic Web, whose layer structure is shown in Figure 1, is one of the most hopeful prospects for further evolution of the Internet. If the Internet is perceived in this manner, then access to knowledge resources will be provided via ontology languages, which makes these resources machine-readable and hence susceptible to further processing by machines, too. In the course of its development, the semantic Web has originated several standards which account for its vast potential, including the XML and XML Schema markup languages; vocabulary, resource or ontology description languages, such as RDF, RDFS, OWL, OWL2; rule languages, such as SWRL, RuleML and RIF, supporting the definition of rules and the transfer of knowledge between different inference mechanisms; and the query language SPARQL, whose structure resembles that of SQL and allows generation of queries based on ontologies expressed in the RDF language.

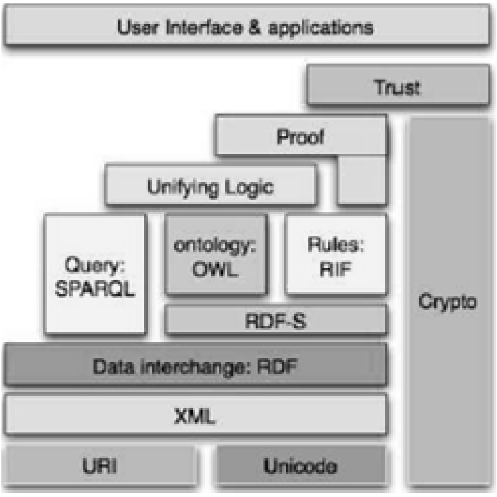


Figure 1. The layer structure of the semantic Internet

Source: Gerber, van der Merwe, Barnard [2008].

The semantic Web has already reached a fairly mature stage of advancement and there are a variety of implementations available across the Internet, including browsers, such as PowerSet, Yahoo! SearchMonkey, Google Rich Snippets, and Hakia, and semantic wikis, such as Semantic MediaWiki, BiomedGT, VisWiki.

The application of semantics and the combination of data and metadata allows [Bąk, Jędrzejek 2009]:

- searching information based on meaning (i.e., the meaning of words entered into a browser application) rather than on keywords only, the latter approach producing useless finds and being unable to handle synonyms,
- disambiguation of data based on context,
- extracting information from diverse sources and its automatic integration,
- presenting just the data which is relevant to the user along with an explanation of the actions undertaken and the conclusions delivered,
- reasoning on semantically described data, thus generating new information (not yet explicitly represented) which, if subject to another inference process, may lead to the discovery of further new facts, etc.,
- incorporating knowledge contained in documents of a specific type, which could facilitate the management of documents and knowledge, helping maintain data collections non-contradictory and up-to-date,
- automated generation of documents including semantic descriptions,
- automatic generation of Web pages including semantic information without any input or actions on the user's part (requisite knowledge is derived from the semantics of the underlying ontology).

Nevertheless, the application of ontology within the semantic Web involves the prior development of mechanisms for its creation and transformation. It is much simpler to use microformats instead even though their power and functionality are inferior.

2.2. Microformats

Microformats, being an extension to the existing Web page design and construction methods, do not constitute a new standard for representing knowledge resources. Rather, they are supposed to establish a convention for extending the *class*, *alt* and *title* markups by a number of keywords or names making up a specific microformat. These keywords or names can be then used to make it easier to find relevant data on a website. One such microformat is RDFa, inscribed in the W3C standard since 2008. This microformat is similar to the RF ontology language, except that it employs attributes instead of tags to attach labels describing Web content.

Items of information thus embedded in a Web page are referred to as elements and have certain properties. For example, for an element "person", properties such as "name", "nickname", "gender", etc., can be defined. The following is an example of how Web content can be enriched with this sort of tags:

```
<div xmlns:v=”http://rdf.data-vocabulary.org/#” typeof=”v:Person”>

   Nazwisko i imie:<span property=”v:name”>Mariusz Żytniewski</span>,
   </BR>
   Pseudonim: <span property=”v:nickname”>Zyto</span>,</BR>
   strona domowa:
   <a href=”http://zyto.netmark.pl” rel=”v:url”>zyton.netmark.pl</a>.

</div>
```

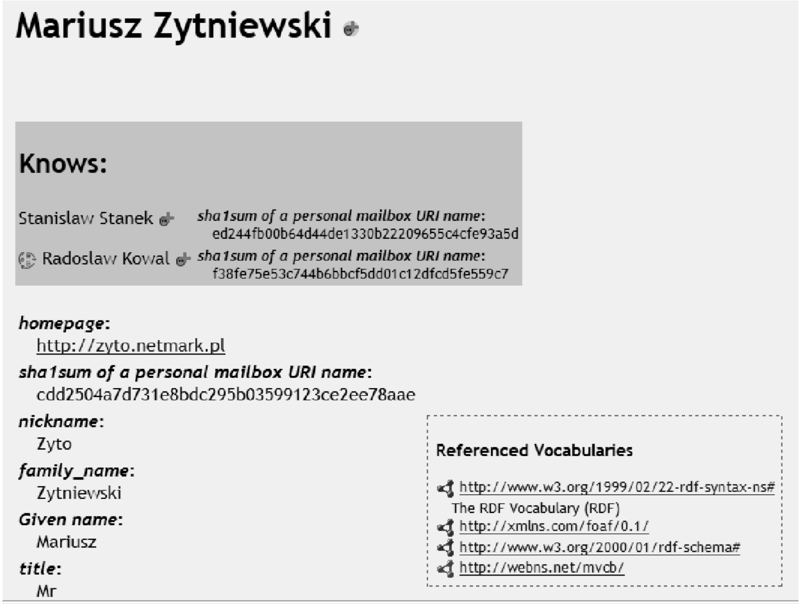
In the above example, the namespace http://rdf.data-vocabulary.org/# is used, which provides a context for defining persons, places, products and opinions. The attribute typeof=”v:Person” indicates that the markups will designate a person. Further in the code (the example only reveals a fragment comprised in a single DIV block), one can apply other “property” attributes to specify further properties. Data which has been included in this way on a webpage can be now easily processed by a wide range of software applications. Google robots, for example, can handle such a markup [*Introducing Rich Snippets* 2012].

Another microformat which can be used to describe data is FOAF (Friend of a Friend), permitting the definition of person’s properties. However, it operates on slightly different principles. With this type of microformat, one needs to create a foaf.rdf file containing the pre-processed information to be published and, subsequently, include it in the HEAD section. The file can be generated, for example, using an automated tool available at http://www.ldodds.com/foaf/foaf-a-matic. The resulting file may look as follows:

```
<rdf:RDF
      xmlns:rdf=”http://www.w3.org/1999/02/22-rdf-syntax-ns#”
      xmlns:rdfs=”http://www.w3.org/2000/01/rdf-schema#”
      xmlns:foaf=”http://xmlns.com/foaf/0.1/”
      xmlns:admin=”http://webns.net/mvcb/”>
<foaf:PersonalProfileDocument rdf:about=””>
   <foaf:maker rdf:resource=”#me”/>
   <foaf:primaryTopic rdf:resource=”#me”/>
   <admin:generatorAgent rdf:resource=”http://www.ldodds.com/foaf/
   foaf-a-matic”/>
   <admin:errorReportsTo rdf:resource=”mailto:leigh@ldodds.com”/>
</foaf:PersonalProfileDocument>
<foaf:Person rdf:ID=”me”>
<foaf:name>Mariusz Żytniewski</foaf:name>
<foaf:title>Mr</foaf:title>
<foaf:givenname>Mariusz</foaf:givenname>
<foaf:family_name>Żytniewski</foaf:family_name>
<foaf:nick>Zyto</foaf:nick>
<foaf:mbox_sha1sum>cdd2504a7d731e8bdc295b03599123ce2ee78aae</
foaf:mbox_sha1sum>
<foaf:homepage rdf:resource=”http://zyto.netmark.pl”/>
<foaf:knows>
<foaf:Person>
<foaf:name>Radosław Kowal</foaf:name>
<foaf:mbox_sha1sum>f38fe75e53c744b6bbcf5dd01c12dfcd5fe559c7</
foaf:mbox_sha1sum>
<rdfs:seeAlso rdf:resource=”http://link-do-innego-pliku-foaf”/></
foaf:Person></foaf:knows>
<foaf:knows>
<foaf:Person>
<foaf:name>Stanisław Stanek</foaf:name>
```

```
<foaf:mbox_sha1sum>ed244fb00b64d44de1330b22209655c4cfe93a5d</
foaf:mbox_sha1sum></foaf:Person></foaf:knows></foaf:Person>
</rdf:RDF>
```

With a description so defined, links between persons can be easily specified, and other files with FOAF definitions can be referenced, too. A sample FOAF definition is shown in Figure 2.



Mariusz Zytniewski

Knows:

Stanislaw Stanek — *sha1sum of a personal mailbox URI name*: ed244fb00b64d44de1330b22209655c4cfe93a5d

Radoslaw Kowal — *sha1sum of a personal mailbox URI name*: f38fe75e53c744b6bbcf5dd01c12dfcd5fe559c7

homepage:
http://zyto.netmark.pl

sha1sum of a personal mailbox URI name:
cdd2504a7d731e8bdc295b03599123ce2ee78aae

nickname:
Zyto

family_name:
Zytniewski

Given name:
Mariusz

title:
Mr

Referenced Vocabularies

http://www.w3.org/1999/02/22-rdf-syntax-ns#
The RDF Vocabulary (RDF)
http://xmlns.com/foaf/0.1/
http://www.w3.org/2000/01/rdf-schema#
http://webns.net/mvcb/

Figure 2. A sample definition read from a FOAF file

Source: authors' own study.

Such microformats can be successfully utilized in developing software agents. A library which supports the processing of semantically represented knowledge resources and is often deployed in building inference engines for use by software agents is JENA.

2.3. Semantic knowledge processing – the example of JENA

To create one's own FOAF file using JENA, all one needs to do is to indicate a FOAF namespace by importing the library com.hp.hpl.jena.sparql.vocabulary.FOAF. As a result, you will be able to employ the relevant properties, e.g., FOAF.type or FOAF.title. The lines of code reproduced below demonstrate how the JENA library permits you to create your own FOAF file:

```
public class Mikroformaty {

   public static void main(String[] args) {

   Model model = ModelFactory.createDefaultModel();
model.setNsPrefix("foaf", FOAF.NS);
Resource contributor = model.createResource("http://zyto.netmark.
pl");
contributor.addProperty(RDF.type, FOAF.Person);
contributor.addProperty(FOAF.title, "Mr");
contributor.addProperty(FOAF.name, "Mariusz Zytniewski");
model.write(System.out, "RDF/XML-ABBREV");

   }
```

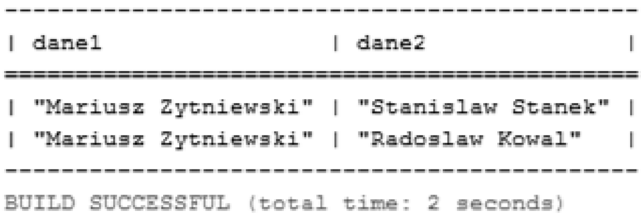
To read the file, anyone knowing its URL can make a connection and retrieve it using the URL, URLConnection and InputStream classes.

```
public static void main(String[] args) {
String u = "http://zyto.netmark.pl/zyto.foaf";

URL url = new URL(u);
URLConnection urlconnection = url.openConnection();
urlconnection.connect();

java.io.InputStream in = urlconnection.getInputStream();
model.read(in, u);
model.write(System.out, "RDF/XML-ABBREV");

in.close();

}
```

With a markup like the one shown above, software agents can attempt to perform inferences. The SPARQL language can be used for this purpose. In the example below SPARQL is applied to establish links among individuals.

```
String queryString =
   "PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>"+
   "PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema>"+
   "PREFIX foaf:<http://xmlns.com/foaf/0.1/>"+
   "PREFIX admin:<http://webns.net/mvcb/>"+
"SELECT ?dane1 ?dane2 " +
"WHERE { "+
   " ?osoba1 foaf:knows ?osoba2 ."+
   " ?osoba1 foaf:name ?dane1 ."+
   " ?osoba2 foaf:name ?dane2 ."+
   " }";

com.hp.hpl.jena.query.Query query = QueryFactory.
create(queryString);
QueryExecution qe = QueryExecutionFactory.create(query, model);
com.hp.hpl.jena.query.ResultSet results = qe.execSelect();
ResultSetFormatter.out(System.out, results, query);
qe.close();
```

When executed, the code will output what follows (see Figure 3):



```
---------------------------------------------
| dane1                 | dane2               |
=============================================
| "Mariusz Zytniewski" | "Stanislaw Stanek" |
| "Mariusz Zytniewski" | "Radoslaw Kowal"   |
---------------------------------------------
BUILD SUCCESSFUL (total time: 2 seconds)
```

Figure 3. The outcome of SPARQL code executed on a FOAF file

Source: authors' own study.

3. Conclusions

Arguably, the content of Web 2.0 and they way it conveys information are user-centered, i.e., users are both producers and recipients of information circulated via the medium. The authors believe that, as long as this is the case, technologies that are in use today are perfectly adequate. However, where content is to be increasingly often processed by computers, appropriate standards for describing Internet resources will be required.

This paper, devoted to a discussion of issues relating to semantic data processing and microformats, aims to demonstrate that under Web 3.0 such a standard is offered by ontology description languages. Microformats, on the other hand, appear to be a solution which combines both the approaches, at once facilitating semantic description of WWW content and its further machine-processing. However, although they do make it possible to automate the processing of Web content by dedicated software, their functionality is inferior to that of ontologies.

References

Bąk J., Jędrzejek Cz., *Semantic Web – technologie, zastosowania, rozwój*, PLOUG'09. Politechnika Poznańska, Poznań 2009.

Fasli M., *Agent Technology for e-Commerce*, John Wiley & Sons, 2007.

Gerber A., van der Merwe A., Barnard A., A functional semantic web architecture, [in:] *Proceedings of the 5th European Semantic Web Conference (ESWC'08)*, Tenerife Spain, 1-5 June 2008.

Introducing Rich Snippets, http://googlewebmastercentral.blogspot.com/2009/05/introducing-rich-snippets.html (accessed: 1.01.2012).

Luck M., Ashri R., d'Inverno M., *Agent-Based Software Development (Agent-Oriented Systems)*, Artech House Publishers, 2004.

Stanek S., *Modele hybrydowe w podejmowaniu finansowych decyzji gospodarczych*, Wydawnictwo Akademii Ekonomicznej w Katowicach, Katowice 2007.

Zieliński Z., Społeczeństwo informacyjne w dobie Web 2.0, [in:] *Biblioteka w społeczeństwie informacyjnym*, Wyższa Szkoła Handlowa, Kielce 2008, http://www.elearningonline.pl/wp-content/upload/konf_wsh08.pdf (accessed: 1.01.2012).

ZASTOSOWANIE MIKROFORMATÓW W BUDOWIE AGENTÓW PROGRAMOWYCH

Streszczenie: W niniejszym artykule zaprezentowano kwestie dotyczące budowy semantycznego Internetu oraz możliwości zastosowania mikroformatów w technologiach agentowych. W rozdziale pierwszym omówiono zagadnienia dotyczące społeczeństwa informacyjnego oraz WEB 2.0. W rozdziale drugim ukazano koncepcje semantycznego Internetu i mikroformatów oraz przykład zastosowania biblioteki JENA.

Słowa kluczowe: semantyczny Internet, mikroformaty, agent programowy.